FMUOS

Primjeri ispitnih zadataka

7.4.2014.

Teorija i definicije iz formalne logike

1. Osnovna pravila ekvivalencije (pazi na distribuciju !)
2. Semantika kao interpretacija i evaluacija.
3. Definicija "Formalan logički sustav".
4. Osnovna pravila te "Modus ponens" i "Modus Tolens".
5. Definicije u Formalnom logičkom sustavu (teorem/dokaz, konzistentan, odrediv, poluodrediv, model, zadovoljivost, tautologija, logička posljedica, ispravan, kompletan).
6. Definicija ekvivalencije preko logičke posljedice.
7. Normalni oblici (DNF, CNF, literal, klauzula).
8. Konverzija propozicijske formule u CNF oblik. $((A \Rightarrow B) \Rightarrow C) =$

   $= (\neg(\neg A \vee B) \vee C) = ((A \wedge \neg B) \vee C) = (A \vee C) \wedge (\neg B \vee C)$
9. SAT problem.
10. Teorem dedukcije.
11. Izvođenje logičke posljedice uporabom SAT rješavača.

"Tkogod ima majku ima i oca."

M(x, y) – x je majka od y

O(x, y) – x je otac od y

Uz primjenu predikata =(x, y), čija je semantika x jednako y, preslikaj rečenicu u formalizam predikatne logike:

"Ante ima barem dvije sestre."

Ante – konstanta

sestra(x, y) – x je sestra od y

"Tkogod ima majku ima i oca."

M(x, y) – x je majka od y

O(x, y) – x je otac od y

$$\forall x\ [(\exists y\ M(y, x) \Rightarrow \exists z\ O(z, x))]$$

Uz primjenu predikata =(x, y), čija je semantika x jednako y, preslikaj rečenicu u formalizam predikatne logike:

"Ante ima barem dvije sestre."

Ante – konstanta

sestra(x, y) – x je sestra od y

$$\exists x\ \exists y\ [\text{sestra}(x, \text{Ante}) \wedge \text{sestra}(y, \text{Ante}) \wedge \neg{=}(x, y)]$$

„Samo mrtav vanzemaljac je dobar vanzemaljac.“

Mrtav(x) – x je mrtav

Dobar(x) – x je dobar

VZ(x) – x je vanzemaljac

„Za svaki brijeg u Hrvatskoj postoji viši brijeg u Hercegovini.“

Brijeg(x) – x je brijeg

UHrv(x) – x je u Hrvatskoj

UHer(x) – x je u Hercegovini

Viši(x, y) – x je viši od y

„Samo mrtav vanzemaljac je dobar vanzemaljac."

Mrtav(x) – x je mrtav

Dobar(x) – x je dobar

VZ(x) – x je vanzemaljac

$$\forall x\ [(VZ(x) \wedge Mrtav(x)) \Rightarrow Dobar(x)\ ]$$

„Za svaki brijeg u Hrvatskoj postoji viši brijeg u Hercegovini."

Brijeg(x) – x je brijeg

UHrv(x) – x je u Hrvatskoj

UHer(x) – x je u Hercegovini

Viši(x, y) – x je viši od y

$$\forall x\ [(Brijeg(x) \wedge UHrv(x)) \Rightarrow$$
$$\exists y\ \ (Brijeg(y) \wedge UHer(y) \wedge Viši(y, x))$$

„Seviljski brijač brije sve ljude osim onih koji se sami briju.“

Brije(x, y) – x brije y

SB – Seviljski brijač (konstanta)

„U Hrvatskoj postoje najmanje dva nacionalna parka“.

UHrv(x) – x je u Hrvatskoj

NP(x) – x je nacionalni park

Nije_Jednako(x, y) – x i y nisu jednaki

(može $\neq$(x, y) ili bilo što razumljivo za nejednakost)

„Seviljski brijač brije sve ljude osim onih koji se sami briju.“

Brije(x, y) – x brije y

SB – Seviljski brijač (konstanta)

$\forall x\ (\neg brije(x, x) \Rightarrow brije(SB, x))$

„U Hrvatskoj postoje najmanje dva nacionalna parka“.

UHrv(x) – x je u Hrvatskoj

NP(x) – x je nacionalni park

Nije_Jednako(x, y) – x i y nisu jednaki

(može $\neq(x, y)$ ili bilo što razumljivo za nejednakost)

$\exists x\ \exists y\ (NP(x) \wedge UHrv(x) \wedge NP(y) \wedge UHrv(y) \wedge Nije_Jednako(x, y))$

"Svaku štetu plaća neka osoba."

S(x) – x je šteta

O(x) – x je osoba

P(x, y) – x plaća za y

"Postoji točno jedna knjiga na polici."

K(x) – x je knjiga

P(x) – x je na polici

J(x, y) – x je jednako y

"Svaku štetu plaća neka osoba."

$S(x)$ – x je šteta

$O(x)$ – x je osoba

$P(x, y)$ – x plaća za y

$$\forall x[\, S(x) \Rightarrow (\, \exists y\, (O(y) \wedge P(y, x)\,)\,]$$

"Postoji točno jedna knjiga na polici."

$K(x)$ – x je knjiga

$P(x)$ – x je na polici

$J(x, y)$ – x je jednako y

$$\exists x\, [(\, K(x) \wedge P(x)\,) \wedge \forall y\, (\, (K(y) \wedge P(y)) \Rightarrow J(x, y)\,)\,]$$

"Svaki student je mlađi od nekog nastavnika."

S(x) – student

N(x) – nastavnik

M(x, y) – x je mlađi od y

"Svaki sin moga oca je moj brat."

S(x, y) – x je sin od y

O(x, y) – x je otac od y

B(x, y) – x je brat od y

m – ja (konstanta)

"Svaki student je mlađi od nekog nastavnika."

$S(x)$ – student

$N(x)$ – nastavnik

$M(x, y)$ – x je mlađi od y

$$\forall x\, [S(x) \Rightarrow \exists y\, (N(y) \wedge M(x, y))]$$

"Svaki sin moga oca je moj brat."

$S(x, y)$ – x je sin od y

$O(x, y)$ – x je otac od y

$B(x, y)$ – x je brat od y

m – ja (konstanta)

$$\exists x\, \forall y\, [(O(x, m) \wedge S(y, x)) \Rightarrow B(y, m)]$$

"Ne postoji cigla koja je na nekoj cigli ako je ta također na cigli." (pomoć: samo su dvije cigle jedna na drugoj)

na(x, y) – x cigla je na y cigli

"Samo je jedan student pao Formalne metode".

fmuos – konstanta

Student(x) – x je student

Pao(x, fmuos) – x je pao FMUOS

"Ne postoji cigla koja je na nekoj cigli ako je ta također na cigli." (pomoć: samo su dvije cigle jedna na drugoj)

na(x, y) – x cigla je na y cigli

$$\forall x\ \forall y\ [na(x, y) \Rightarrow \neg\exists z\ na(z, x)]$$

Ili

$$\forall x\ \forall y\ [na(x, y) \Rightarrow \neg\exists z\ na(y, z)]$$

"Samo je jedan student pao Formalne metode".

fmuos – konstanta

Student(x) – x je student

Pao(x, fmuos) – x je pao FMUOS

$$\exists x\ (Student(x) \wedge Pao(x, fmuos) \wedge \forall y\ (Student(y) \wedge Pao(y, fmuos) \Rightarrow =(x, y))$$

Sintaksa CTL: označi neispravno definirane

$A[p\ U\ EF \neg r]$

$AEF\ r$

$FG\ r$

$AF[(r\ U\ q) \wedge (p\ U\ r)]$

$A[\neg p\ U\ A[q\ U\ r]]$

$A\neg G\ \neg p$

$AG(q \Rightarrow EG\ r)$

$EF(r\ U\ q)$

$AG\ AF\ r$

$E[A(p1\ U\ p2)\ U\ p3]$

$EF\ EG\ (r \Rightarrow AFp)$

$F(p\ U\ q)$

Sintaksa CTL: označi neispravno definirane

$A[p\ U\ EF\neg r]$

$AEF\ r$

$FG\ r$

$AF[(r\ U\ q) \wedge (p\ U\ r)]$

$A[\neg p\ U\ A[q\ U\ r]]$

$A\neg G\ \neg p$

$AG(q \Rightarrow EG\ r)$

$EF(r\ U\ q)$

$AG\ AF\ r$

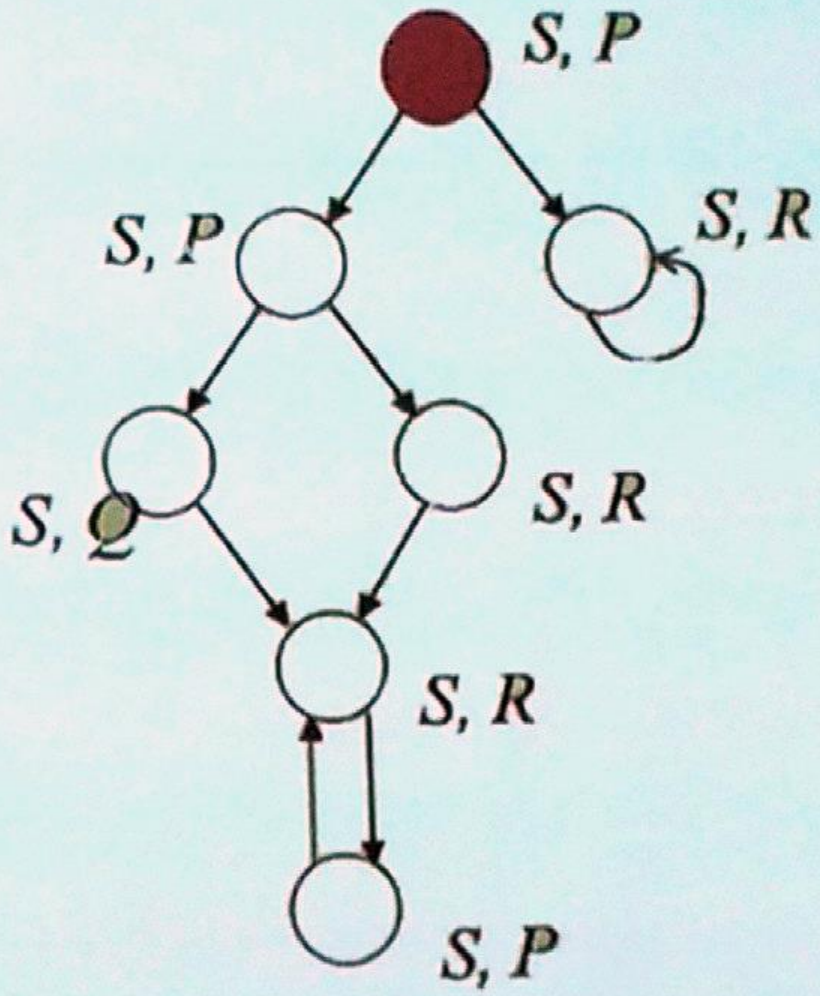
$E[A(p1\ U\ p2)\ U\ p3]$

$EF\ EG\ (r \Rightarrow AFp)$

$F(p\ U\ q)$

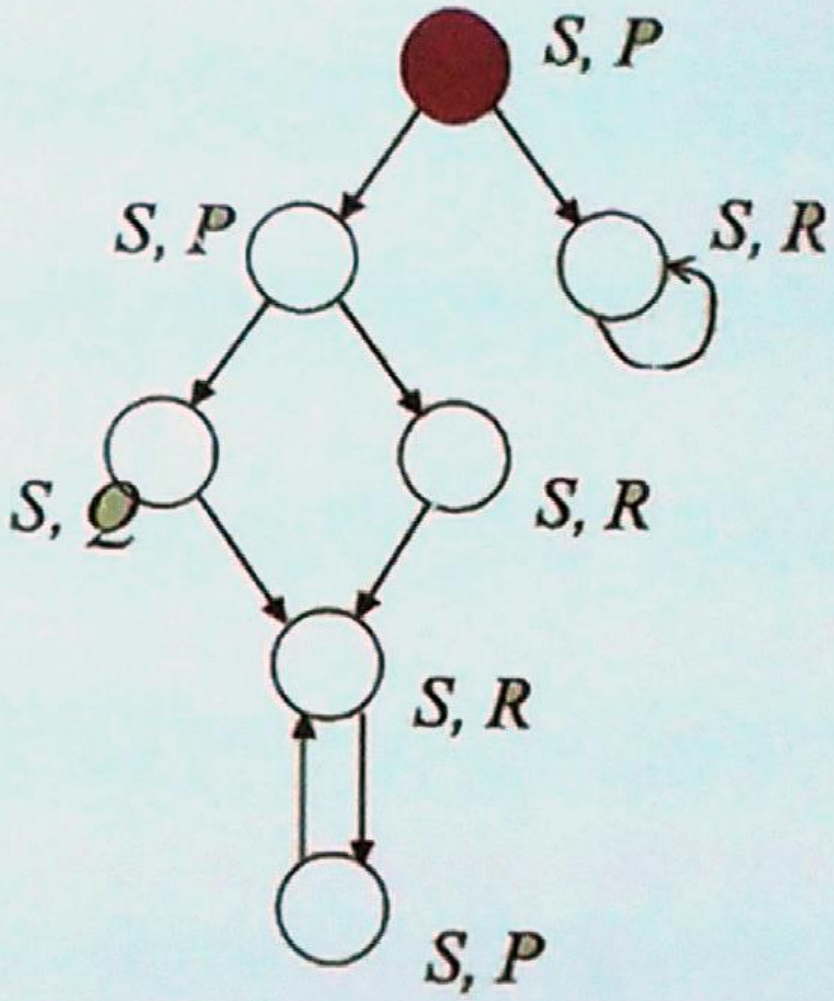
Provjera modela: da li za stanje na vrhu vrijede ili ne CTL formule:

AF Q NE



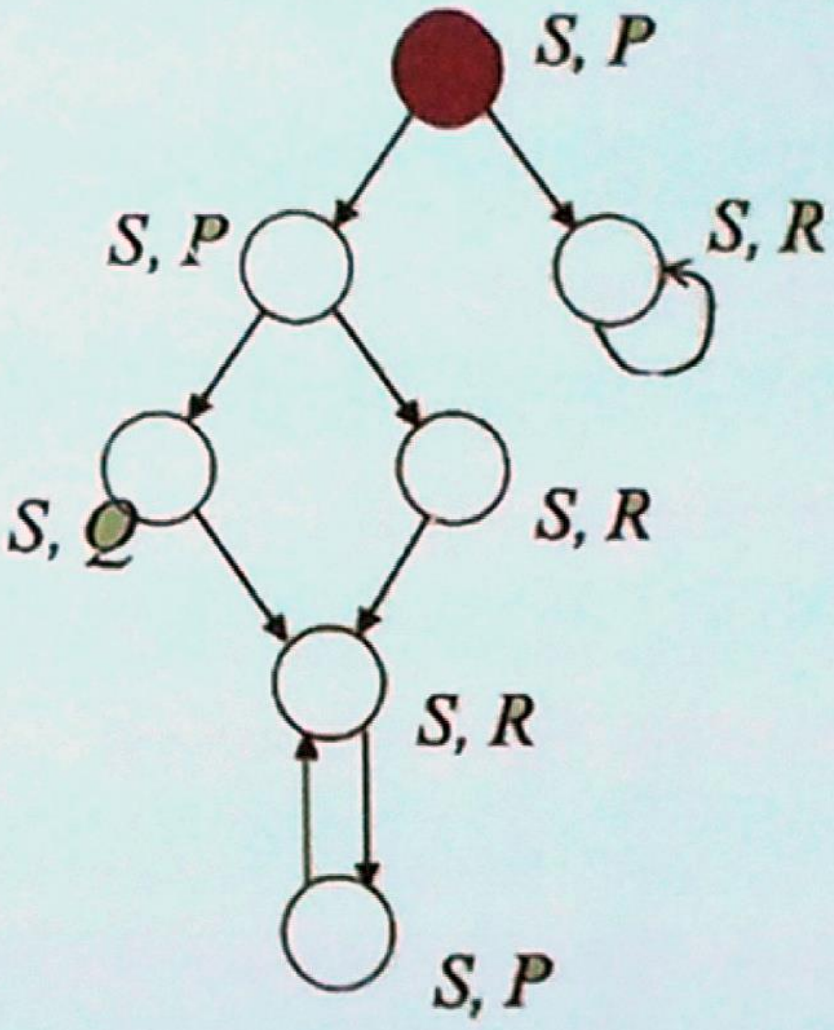
Da li za stanje na vrhu vrijede ili ne CTL formule:

E[P U E(Q U R)] DA



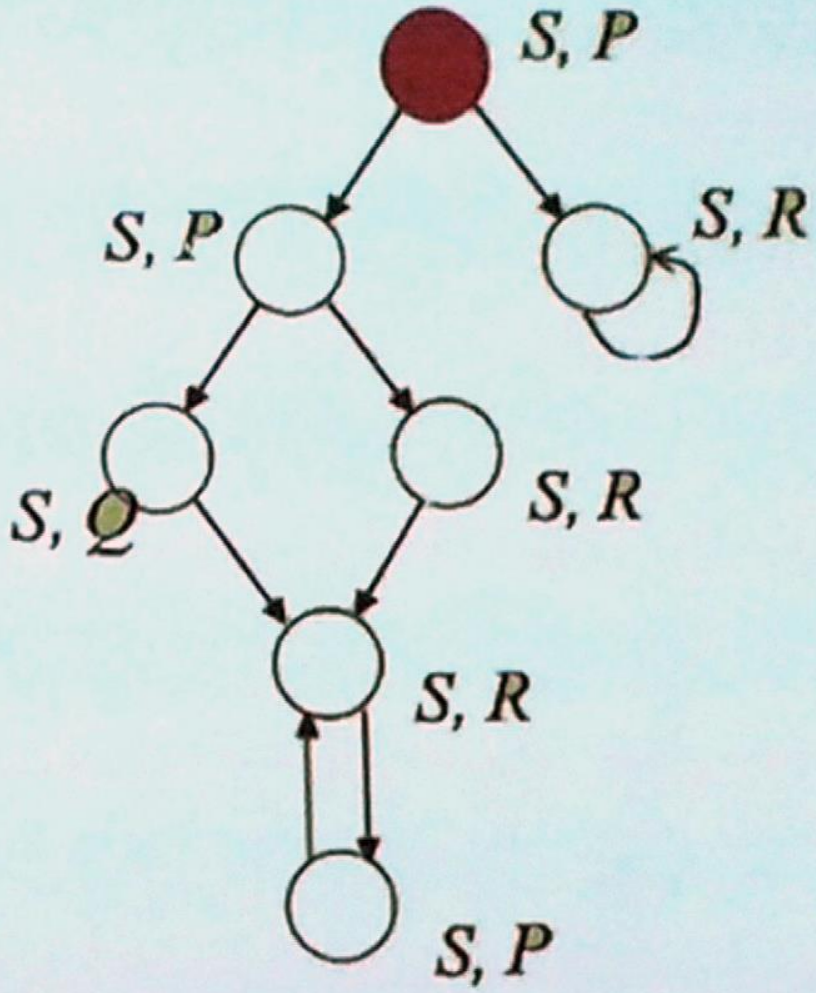
Da li za stanje na vrhu vrijede ili ne CTL formule:

E[P U E(R U Q)] DA



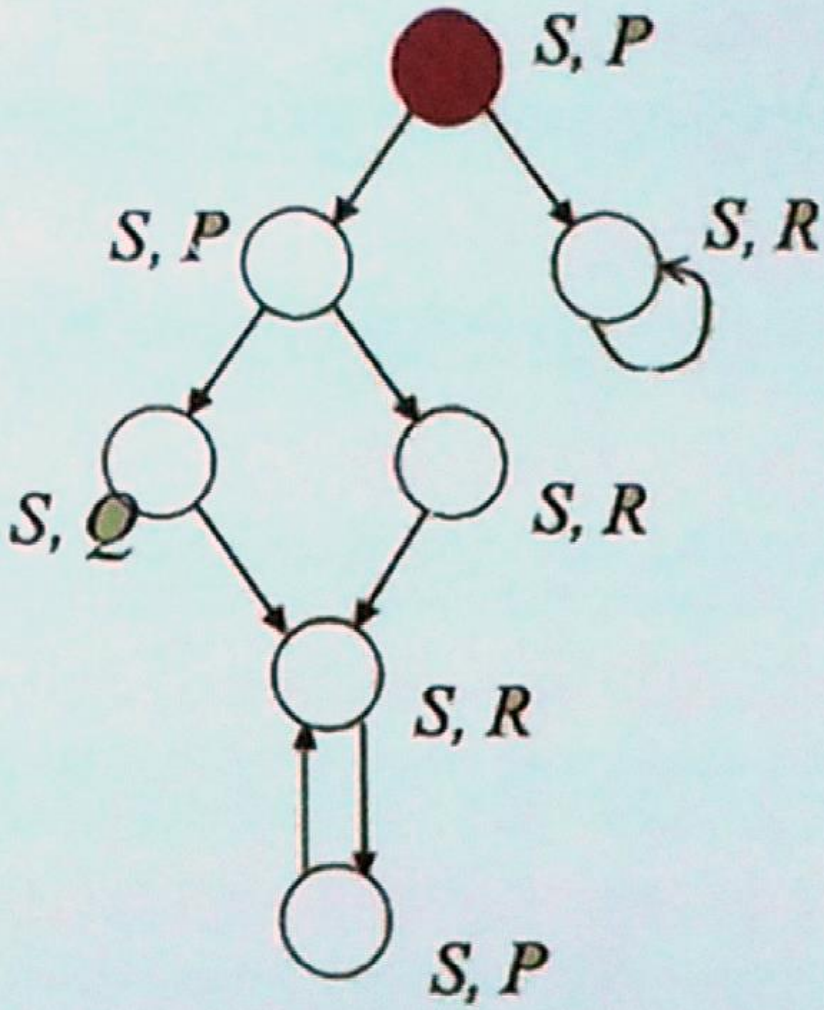
Da li za stanje na vrhu vrijede ili ne CTL formule:

E [E(P U R) U Q] DA

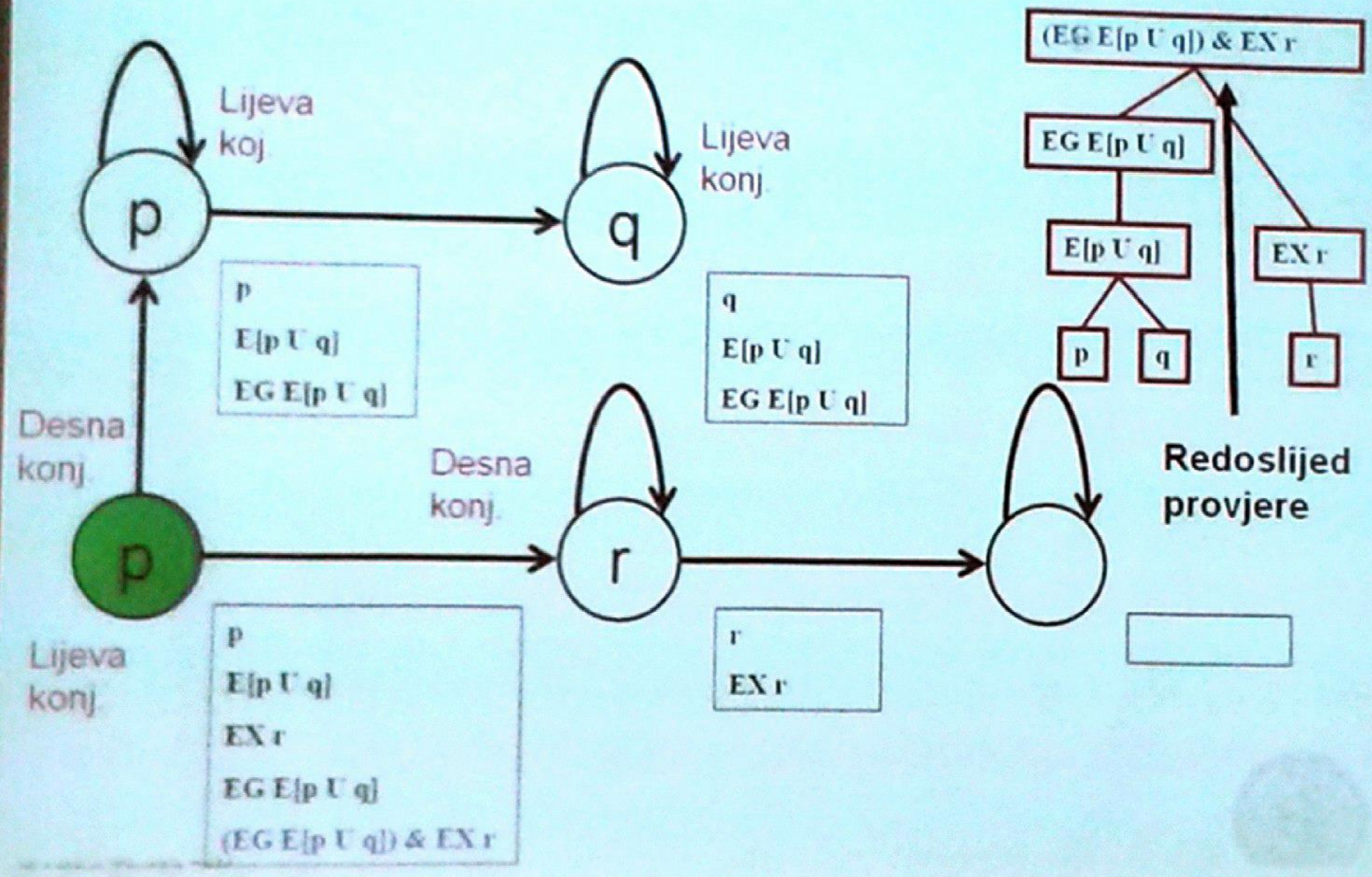


Da li za stanje na vrhu vrijede ili ne CTL formule:

AG EF P	NE
AF R	DA
AG EF R	DA
E[P U E(R U P)]	DA



Pronađi <u>sva stanja</u> za koja vrijedi CTL formula: ((EG E[p U q]) ∧ EX r)



Primjeri specifikacije u CTL-u:

1. "Signali ack1 i ack2 se nikada ne postavljaju istovremeno (simultano)."

2. "Signal ack se nikad ne postavlja (u logičku jedinicu = 1 = TRUE) bez zahtjeva req."

3. Neka su oznake za neki proces:
   n – nije u kritičnoj sekciji
   c – nalaz i se u kritičnoj sekciji
   t – zahtijeva ulazak u kritičnu sekciju

   "Proces može uvijek zahtijevati ulazak u kritičnu sekciju"

Primjeri specifikacije u CTL-u:

1. "Signali ack1 i ack2 se nikada ne postavljaju istovremeno (simultano)."

   $AG[\neg(ack1 \wedge ack2)]$

2. "Signal ack se nikad ne postavlja (u logičku jedinicu = 1 = TRUE) bez zahtjeva req."

   $\neg E(\neg req\ U\ ack)$

   ne postoji put (izvršavanje modela) na kojem bi nakon nepostojanja req, konačno bio ack.
   AG(req U ack) može imati ack u sadašnjem stanju !

3. Neka su oznake za neki proces: n – nije u kritičnoj sekciji
   c – nalaz i se u kritičnoj sekciji
   t – zahtijeva ulazak u kritičnu sekciju

   "Proces može uvijek zahtijevati ulazak u kritičnu sekciju"

   $AG(n \Rightarrow EX\ t)$

Rekapitulacija eksplicitnih metoda izračunavanja CTL formula

Potrebno je pronaći sva stanja za koja je istinita neka formula vremenske logike.

Dovoljno je pronaći postupke za izračunavanje **EX, EG, EU** (jer je to adekvatan set formula).

Izračunavanje EX

1. Zadan je skup stanja u kojima je istinita formula f: Q(f).
2. Tražimo skup stanja za koja je istinito EX f, t.j. Q(EX f).

   To znači da tražimo stanja iz kojih je u jednom koraku moguće doći do nekog stanja (ili više njih) iz zadanog skupa Q(f). Drugim riječima tražimo prethodnike za stanja iz skupa Q(f). To nam daje inverzna relacija zadane Kripke strukture.

$$Q(EX\ f) = R^{-1}Q(f)$$

Izračunavanje EG (1)

1. Zadan je skup stanja u kojima je istinita formula f: Q(f).

2. Tražimo skup stanja za koja je istinito EG f, t.j. Q(EG f).

   To znači da tražimo sva stanja iz kojih postoji bar jedno izvođenje (bar jedan put) na kojem je stalno istinita formula f. Drugim riječima tražimo stanja iz kojih bar na jednom putu počinju i stalno slijede barem neka stanja iz skupa Q(f). Na putu ne moraju slijediti sva stanja iz skupa Q(f) ali moraju biti iz Q(f).

3. Započinjemo s rekurzivnom formulom CTL vremenske logike:

   $EG\ f = f \wedge EX\ EG\ f$

   Za neki put na kojem je istinito EG f, mora biti f istinit u sadašnjem stanju i kad se pomaknemo za jedan korak opet mora biti istinito EG f.

Izračunavanje EG (2)

4. Istu rekurzivnu formulu možemo napisati kao izračun stanja:

$$Q(EG\ f) = Q(f) \cap Q(EX\ EG\ f)$$

Stanja za koja je istinito EG f su neka zadana stanja Q(f) za koja je istinito (logička konjunkcija ili presjek skupova) kada se iz svakog od njih pomaknemo za jedan korak opet je istinito EG f.

5. Supstituiramo rezultat za EX:

$$Q(EG\ f) = Q(f) \cap R^{-1}\ Q(EG\ f)$$

6. Rješenje ove jednadžbe traži teoriju čvrste točke (engl. fix point).

Sve monotone funkcije posjeduju najmanju i najveću čvrstu točku, t.j. X je čvrsta točak akko:

$$F(X) = X$$

U kontekstu izračunavanja skupova X je podskup nekog skupa S, a F je funkcija na svim podskupovima od S.

Izračunavanje EG (3)

7. Za monotone funkcije čvrsta točka se računa (Knaster-Tarski teorem):

   $F^{n+1}(\emptyset)$ je najmanji fix-point od F.

   $F^{n+1}(S)$ je najveći fix-point od F.

   Primjenom funkcije n+1 puta počevši od praznog skupa slijedi najmanja čvrsta točka, a primjenom funkcije n+1 puta počevši s cijelim skupom S slijedi najveća čvrsta točka.

8. U jednadžbi za Q(EG f) označimo funkciju F i skupove Q(EG f) = Z:

   $Q(EG\ f) = Q(f) \cap R^{-1}\ Q(EG\ f)$

   $Q(EG\ f) = F\ [\ Q(EG\ f)\ ]$

   $Z_{k+1} = F\ (\ Z_k\ )$

   $Z_{k+1} = Q(f) \cap R^{-1}\ (\ Z_k\ )$

9. Prema teoremu, iterativno primjenjujemo funkciju F tako da započinjemo s cjeloviti, skupm S (za k=0, Z_0=S) sve dok rezultat ne postane invarijantan, t.j. sve dok ne postane $Z_{k+1} = Z_k$:

Izračunavanje EG (4)

```
2^S EG (CTL f)          ; zadan je skup stanja gdje je f=True
{                       ; algoritam vraća skup stanja (kao
                        ; podskup od S) za koja EG f = True
   k:=0; Z_k:=S;
   do {
      Z_{k+1}:=Q(f) ∩ R^{-1}(Z_k);
      if (Z_{k+1} = Z_k) return Z_k;
      k++;
      } forever;
}
```

Izračunavanje EU (1)

1. Zadan je skup stanja Q(f) u kojima je istinito f i skup stanja Q(g) u kojime je istinito g.

2. Tražimo skup stanja Q[E(f U g)], t.j. stanja za koja je istinito E(f U g) .

   To znači da tražimo stanja Q(g) ali i neka iz Q(f) iz kojih će nakon jednog koraka iz njih slijediti stanja za koja je istinito E(f U g).

3. Započinjemo s rekurzivnom formulom CTL vremenske logike:

   $$E(f\ U\ g) = g \vee (f \wedge EX\ E(f\ U\ g))$$

   Na jednom putu ili je g istinit u sadašnjem stanju ili mora f biti istinit i kada se pomaknemo za jedan korak opet mora biti istinito E(f U g).

4. Istu rekurzivnu formulu možemo napisati kao izračuna stanja:

   $$Q[E(f\ U\ g)] = Q(g) \cup [Q(f) \cap (EX\ E(f\ U\ g))]$$

   Obratiti pažnju na "$\cap$" koji govori da ne sva stanja Q(f) nego samo neka za koja vrijedi EX E(f U g).

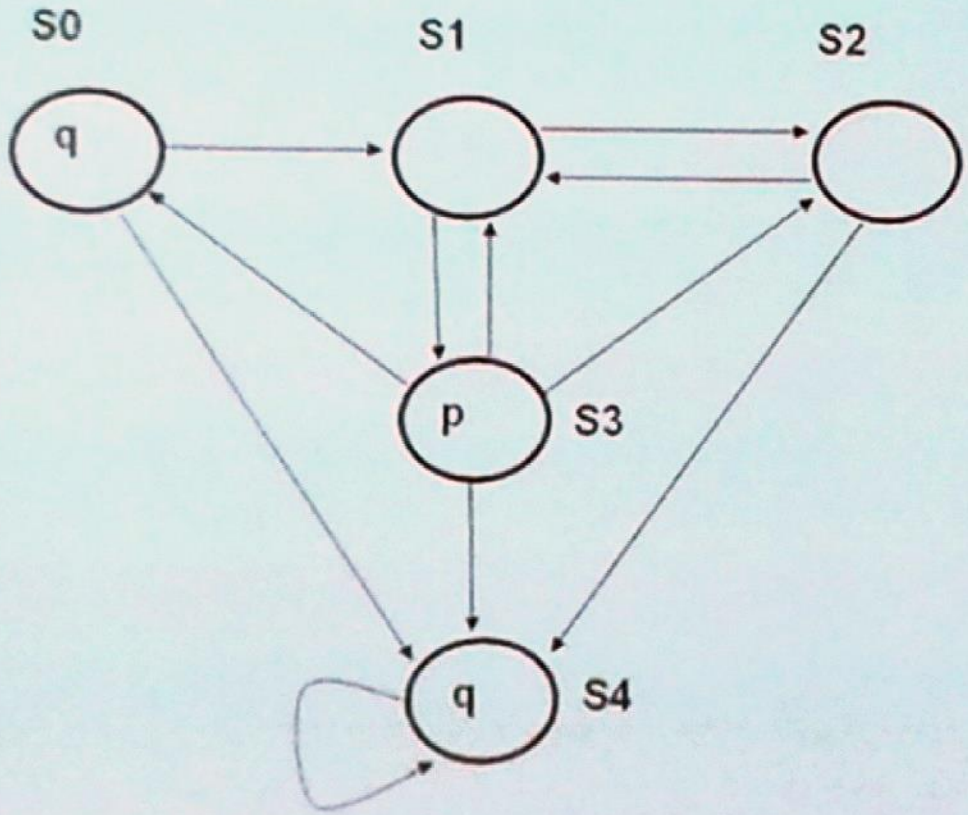
Izračunavanje EU (3)

2^S EU (CTL f, CTL g) ; zadan je skup stanja gdje je
; f=True i skup gdje je g=True
; algoritam vraća skup stanja za
; koja je E(f U g) istinito

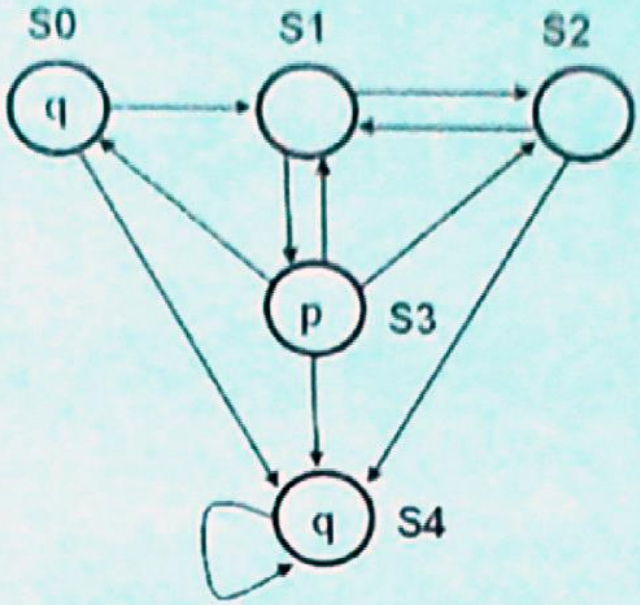
```
{
   k:=0;  Z_0:=∅ ;
   do {
        Z_{k+1} :=Q( g ) ∪ (Q( f ) ∩ R^{-1} ( Z_k ) );
        if (Z_{k+1} =Z_k ) return Z_k ;
        k++;
   } forever;
}
```

Primjeri eksplicitnog izračunavanje stanja koja zadovoljavaju CTL formule vremenske logike

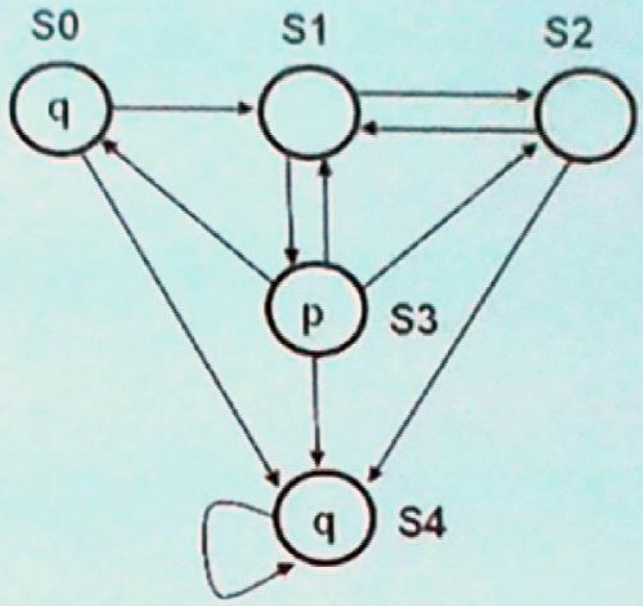
Zadan je model implementacije sustava:



Odredi stanja za koja vrijedi: EF p



Odredi stanja za koja vrijedi: EF p



$EF\ p = E(True\ U\ p)$

$Z_{k+1} = Q(p) \cup [Q(True) \cap R^{-1} Z_k]$

$Z_{k+1} = Q(p) \cup R^{-1} Z_k$

$Q(p) = \{ S3 \}$

$Z_0 = \emptyset$

$Z_1 = \{ S3 \} \cup R^{-1} Z_0 = \{ S3 \} \cup \emptyset = \{ S3 \}$

$Z_2 = \{ S3 \} \cup R^{-1} Z_1 = \{ S3 \} \cup R^{-1} \{ S3 \} = \{ S3 \} \cup \{ S1 \} = \{ S1, S3 \}$

$Z_3 = \{ S3 \} \cup R^{-1} Z_2 = \{ S3 \} \cup R^{-1} \{ S1, S3 \}$

$= \{ S3 \} \cup \{ S0, S1, S2, S3 \}$

$= \{ S0, S1, S2, S3 \}$

$Z_4 = \{ S3 \} \cup R^{-1} Z_3 = \{ S3 \} \cup R^{-1} \{ S0, S1, S2, S3 \}$

$= \{ S3 \} \cup \{ S0, S1, S2, S3 \}$

$= \{ S0, S1, S2, S3 \}$

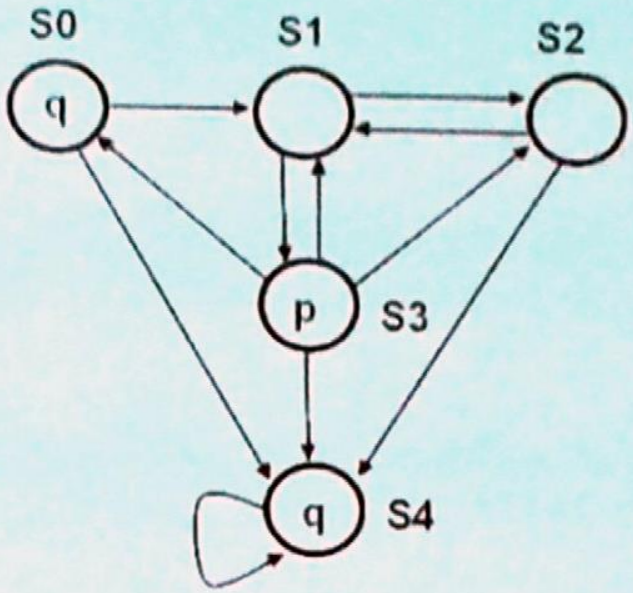
$Z4 = Z3 \Rightarrow$ **FIKSNA TOČKA !**

Rješenje: $Q(EF\ p) = \{ S0, S1, S2, S3 \}$

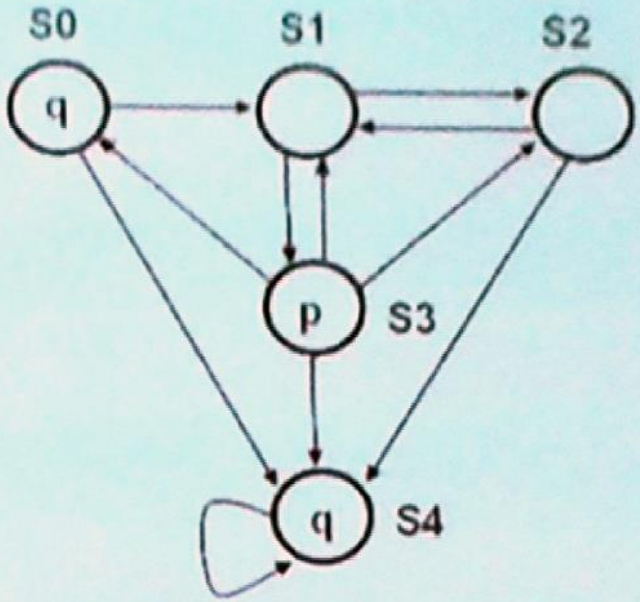
Odredi stanja za koja vrijedi: EG q

$Q(EG\ q) = Q(q) \cap Q(EX\ EG\ q)$

$Z_{k+1} = Q(p) \cap R^{-1} Z_k$



Odredi stanja za koja vrijedi: EG q



$Q(EG\ q) = Q(q) \cap Q(EX\ EG\ q)$

$Z_{k+1} = Q(p) \cap R^{-1} Z_k$

$Q(q) = \{ S0, S4 \}$
$Z_0 = S$ – sva stanja

$Z_1 = \{ S0, S4 \} \cap R^{-1} Z_0 \quad = \{ S0. S4 \} \cap S = \{ S0, S4 \}$

$Z_2 = \{ S0, S4 \} \cap R^{-1} Z_1 \quad = \{ S0, S4 \} \cap R^{-1} \{ S0, S4 \}$
$= \{ S0, S4 \} \cap \{ S0, S2, S3, S4 \}$
$= \{ S0, S4 \}$

$Z_2 = Z_1 \quad \Rightarrow$ **FIKSNA TOČKA !**

Rješenje: $Q\ (EG\ q) = \{ S0, S4 \}$